CSE/PC/B/S/314 Computer Networks Lab

CO2: Design and implement flow control mechanisms of Logical Link Control of Data Link Layer within a simulated network environment.

Assignment 2: Implement flow control mechanisms for Data Link Layer.

Code demonstration due on: 01-05 September 2025 (in your respective lab classes)
Report submission due on: 08-12 September 2025 (in respective lab classes)
Softcopy of the report to be uploaded in the drive.

Language: You can write the program in any high level language like C, C++, Java, Python etc.
Please note that you may need to use these schemes separately for other applications (assignments).

Sender Program: The Sender program consist of following methods:

- **Framing():** This method will prepare the frame following the structure given below. In the header section, the MAC address of the source and destination are specified. Payload is the data of fixed size (pre decided value within the range 46-1500 bytes e.g., 46 bytes) from the input text file. Frame check Sequence using CRC/Checksum (using the CRC/Checksum module of assignment 1) is appended as a trailer.
- **Channel():** The channel method introduces random delay (this will cause packet loss or timeout) and/or bit error (using the error injection module of assignment 1) while transferring frames.

Header <Source Address (6 bytes), Destination Address (6 bytes), Length (2 bytes), Frame seq. no. (1 byte)> 12 bytes	Data <Payload> 46–1500 bytes	Trailer FCS (Frame Check Sequence) <CRC/Checksum> 4 bytes

Fig 1: Data Frame Structure

- **Send():** Sender program will send/transmit data frame using socket connection to Receiver program. Sender should decide whether to send a new data frame or retransmit the same frame again due to timeout.
- **Timer():** Timer will be associated with each frame transmission. It will be used to check the timeout condition.
- **Timeout():** This function should be called to compute the most recent data frame's round-trip time and then re-compute the value of timeout.
- **Recv():** This method is invoked by the sender program whenever an ACK packet is received. Need to consider network time when the ACK was received to check the timeout condition.

Receiver Program: The Receiver program consist of following methods:

- **Recv():** This method is invoked by the Receiver program whenever a Data frame is received.
- **Check():** This method checks (using CRC/Checksum of assignment 1) if there is any error in data. The data frame is discarded if an error is detected otherwise accepted.
- **Send():** Receiver program will prepare an acknowledgement frame and send it to the sender as a response to successful receipt of the data frame.

FlowControl: implement the following flow control approaches using the Sender and Receiver program and their methods:

- **Stop and Wait:** The Sender program calls **Send()** method to transmit one frame at a time slot to the receiver. After that Sender will wait for the ACK frame from the receiver as a response to a successful receipt of the data frame. The interval between sending a message and receiving an acknowledgment is known as the sender's waiting time, and the sender is idle during this time. After receiving an ACK, the sender will send the next data packet to the receiver, and so on, as long as the sender has data to send. If the data frame is lost due to delay or discarded due to error no acknowledgement will be transmitted from the receiver and thus there will be timeout. After timeout, Sender will retransmit the same packet again.

- **Go-Back-N ARQ:** The **Send()** method of the Sender program will take **N** as input. The Sender's window size is N. So that the sender can send N frames which are equal to the window size. The size of the receiver's window is 1. Once the entire window is sent, the sender then waits for a cumulative acknowledgement to send more packets. That is, receiving acknowledgment for frame i means the frames i-1, i-2, and so on are acknowledged as well. You can specify that acknowledgement no. in the ACK frame. Receiver receives only in-order packets and discards out-of-order frames and corrupted frames (checked by CRC or Checksum). In case of packet loss, the entire window would be re-transmitted.

- **Selective Repeat ARQ:** The **Send()** method of the Sender program will take **N** as input. The Sender's window size and Receiver's window size both are N. The sender sends N frames and the receiver acknowledges all frames whether they were received in order or not. Remember that type of acknowledgement here is not cumulative. It uses Independent Acknowledgement to acknowledge the packets. In this case, the receiver maintains a buffer to contain out-of-order packets and sorts them. Receiver discards only corrupted frames (checked by CRC or Checksum). The sender selectively re-transmits the lost packet and moves the window forward. That means Sender retransmits unacknowledged packets after a timeout or upon a NAK (if NAK is employed).

Test the above three schemes for the following cases (not limited to).
- Compare time between the propagation of a packet and reception of its ACK.
- Compare **efficiency** of the above approaches without error or lost frame.
- Compare **efficiency** of the above approaches for different probability (0.1 - 0.5) of an error or delay in the transmission of a packet or in its acknowledgment.